

IN 33-CR

186 576

25P

Communications and Control for Electric Power Systems:

Power Flow Classification for Static Security Assessment

D. Niebur

A. Germond

(NASA-CR-194512) COMMUNICATIONS
AND CONTROL FOR ELECTRIC POWER
SYSTEMS: POWER FLOW CLASSIFICATION
FOR STATIC SECURITY ASSESSMENT
(JPL) 25 p

N94-13948

Unclass

G3/33 0186576

February 1993

Prepared for

Office of Energy Management Systems
United States Department of Energy

Through an agreement with

National Aeronautics and
Space Administration

by

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

1. Report No. 93-6		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Communications and Control for Electric Power Systems: Power Flow Classification for Static Security Assessment				5. Report Date February 1993	
				6. Performing Organization Code	
7. Author(s) D. Niebur A. Germond				8. Performing Organization Report No.	
9. Performing Organization Name and Address JET PROPULSION LABORATORY California Institute of Technology 4800 Oak Grove Drive Pasadena, California 91109				10. Work Unit No.	
				11. Contract or Grant No. NAS7-918	
				13. Type of Report and Period Covered JPL Publication	
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D.C. 20546				14. Sponsoring Agency Code RE152 PL-778-12-01-02-03	
15. Supplementary Notes					
<p>16. Abstract</p> <p>This report investigates the classification of power system states using an artificial neural network model, Kohonen's self-organizing feature map. The ultimate goal of this classification is to assess power system static security in real time. Kohonen's self-organizing feature map is an unsupervised neural network which maps N-dimensional input vectors to an array of M neurons. After learning, the synaptic weight vectors exhibit a topological organization which represents the relationship between the vectors of the training set. This learning is unsupervised, which means that the number and size of the classes are not specified beforehand. In the application developed in this report, the input vectors used as the training set are generated by off-line load-flow simulations. The learning algorithm and the results of the organization are discussed.</p>					
17. Key Words (Selected by Author(s))			18. Distribution Statement Unclassified; unlimited		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 25	
				22. Price	

Communications and Control for Electric Power Systems:

Power Flow Classification for Static Security Assessment

D. Niebur
A. Germond

February 1993

Prepared for

Office of Energy Management Systems
United States Department of Energy

Through an agreement with

National Aeronautics and
Space Administration

by

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was started at the Swiss Federal Institute of Technology, Lausanne (EPFL), sponsored by EPFL, and was completed at the Jet Propulsion Laboratory, California Institute of Technology, sponsored by EPFL. Publication support was provided by the Jet Propulsion Laboratory, sponsored by the U.S. Department of Energy through an agreement with the National Aeronautics and Space Administration.

Reference herein to any commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

THIS REPORT investigates the classification of power system states using an artificial neural network model, Kohonen's self-organizing feature map. The ultimate goal of this classification is to assess power system static security in real-time.

Kohonen's self-organizing feature map is an unsupervised neural network which maps N -dimensional input vectors to an array of M neurons. After learning, the synaptic weight vectors exhibit a topological organization which represents the relationship between the vectors of the training set. This learning is unsupervised, which means that the number and size of the classes are not specified beforehand.

In the application developed in this report, the input vectors used as the training set are generated by off-line load-flow simulations. The learning algorithm and the results of the organization are discussed.

PRECEDING PAGE BLANK NOT FILMED

11 INTENTUALLY BLANK

FOREWORD

THIS REPORT summarizes work done by Dagmar Niebur during 1991 while she was a member of my group at the Jet Propulsion Laboratory (JPL). Dagmar was (and is) a Ph.D. student at the Swiss Federal Institute of Technology at Lausanne. Her coauthor for this report is her faculty advisor, Alain Germond.

The work described here was started while Dagmar was in Switzerland, as part of her thesis work, and continued when she joined my group in 1990. As such, it represented only the second time we had devoted some effort to the topic of artificial intelligence (AI). Our endeavors mirror changes in the field in general, and are worth reviewing.

Some years ago, my predecessor Ralph Caldwell created a power system simulator with the object of demonstrating the difficulties of operating a power system containing renewable-resource generators. It is usually not possible to schedule these sources, and this, combined with the time lags associated with controlling conventional sources, can make for unexpected area control errors. The simulator was so good that, without considerable training, most users experienced difficulties even without the renewable sources. Since the simulator required an expert to run it, it seemed that here was an application for an expert system. Accordingly, Dr. K.A. Bahrami and I developed an expert system to advise the simulator operator. (Expert systems were the first developments of artificial intelligence to achieve widespread popularity. They use a piece of software called an inference engine to apply rules developed by human experts.) We found that even normal operation of the simulator (which was fairly predictable) required a large number of rules. Operation outside a small range close to normal was beyond our abilities. In the end, our simulator and the expert system we developed with it were transferred to Dr. Gerry Sheblé, then a Professor of Power Engineering at Auburn University. He was able to use the simulator as a teaching tool, and further development of the expert system provided material for several student projects. For a while we abandoned work on AI.

Few AI researchers would agree with John Locke, the 17th-century philosopher who thought that the human mind began as a blank slate, to be filled by experience. Most would argue that something in the way of knowledge is needed in order to be able to learn. Over the past few years, a growing number of researchers have investigated the way the neurons of the brain are connected. Perhaps these connections facilitate learning—certainly they represent a highly parallel, distributed way of computing that is quite unlike ordinary digital computers. These networks can be built in hardware, or simulated in software. Both go by the name of artificial neural networks, or neural nets.

One fascinating—and I think suggestive—aspect of neural nets is that their inner workings are not directly controlled, or even accessible. A net may be comprised of dozens or hundreds of simple processors that, like brain neurons, can receive and send signals to many

others. Each processor may be called a node or a neuron. The output from a particular node to another particular node depends in some nonlinear way on the combination of its inputs. The weighting of the various inputs can be changed by "training."

Various groups of AI researchers have concentrated on particular implementations of neural networks, and particular ways of training the nets. At the California Institute of Technology—Caltech is the parent organisation of JPL—Professor J.J. Hopfield has given his name to an interconnected network that functions as a content-addressable memory. Weights are downloaded. Other multi-layer networks can be trained by "back-propagation," which functions much like negative feedback. A two-layer network first described by Professor T. Kohonen in 1982 is trained without supervision. It attracts much attention in Europe. It is a Kohonen network that Dagmar used for the work described in this report.

Neural nets have much to offer over conventional computers. They can be faster and more robust, because of the parallel and distributed nature of the computation. They can also generalize, and respond accurately to situations that they have not seen before. It is this aspect of neural nets that makes them particularly attractive in this application.

HAROLD KIRKHAM

PASADENA, CALIFORNIA
FEBRUARY 1993

CONTENTS

LIST OF FIGURES	viii
INTRODUCTION	1
KOHONEN'S SELF-ORGANIZING FEATURE MAP	3
MODELING OF THE SECURITY ANALYSIS PROBLEM	6
INTERPRETATION OF THE RESULTS	9
INFLUENCE OF NEURAL NET SIZE	13
CHOICE OF SECURITY CRITERIA	14
CONCLUSIONS	15
REFERENCES	16

~~SECRET~~ ¹⁴ INTENTIONALLY BLANK

PRECEDING PAGE BLANK NOT FILMED

LIST OF FIGURES

Figure		Page
1	Architecture of Kohonen's self-organizing feature map	3
2	Mapping of 3-dimensional vectors onto a 2-dimensional Kohonen network	4
3	Operating space of a 3-bus 3-line active power system model	6
4	5-bus 7-line power system	7
5	Apparent power in branch North-Lake	9
6	Apparent power in branch North-South	10
7	Active power generation at bus North	10
8	Security boundaries	10
9	5×5 cluster map	11
10	3×3 cluster map	13

SECTION 1

INTRODUCTION

THE STEADY-STATE load flow model is a well accepted approximation for a power system in normal operation where load and generation vary only slightly around a scheduled case. The evaluation of the present case and the impact of possible line or generator outages can be determined by solving numerically the nonlinear load flow equations for all contingencies, or at least for those ranked as important (Ejebe and Wollenberg, 1979). Because of the combinatorial nature of the problem these approaches require a huge amount of computation time for large power systems and therefore their use for on-line monitoring is limited.

Statistical pattern recognition methods have been proposed to determine those features which mainly influence the system behavior (Pang, Prabhakara, El-Abiad and Koivo, 1974). Associative memory techniques and nearest neighbor classification identify similarities between present and former system states (Pao, 1982; Oh, 1986).

More recently, artificial neural networks have been introduced for power system applications; see for example Sobajic and Pao (1989). A detailed overview on the different neural net types used for power system applications is given by Pao (1990).

For static security assessment a multilayer network was trained using back propagation in order to classify load patterns for a given contingency (El-Sharkawi *et al.*, 1990). Contingencies have also been classified with a Hopfield network (Chow *et al.*, 1990), which was "trained" with the linear programming algorithm. A system of neural networks based on the functional link net architecture was studied by Sobajic, Pao, Njo and Dolce (1990). Power system state clustering is performed unsupervised by the ART-type (adaptive resonance theory) learning. A supervised learning algorithm is then used for feature identification.

In power system operation it is unrealistic to expect that all possible cases will be encountered during off-line simulation. The identification of typical system states, the determination of number and size of classes represented by these states, and the construction of a prototype state for each class reflecting the common features of all class members are therefore very difficult. These parameters are usually determined experimentally before training.

Back-propagation networks require the knowledge of all parameters; Hopfield networks need the specification of typical system states and number of classes. Unsupervised trained networks of the ART-type construct class and its prototype autonomously. However, they need the radius specification. None of these networks is capable of forming classification categories whose class members have not been presented during the training phase.

The self-organizing network proposed by Kohonen (1989) solves these shortcomings. The

number of classes is limited only by the number of neurons. Class prototypes are constructed in an unsupervised manner. The size of the classes reflects the statistical distribution of the power system states. The classification is coarse for states rarely encountered during the training phase and fine for often encountered states. The network further forms classes representing a certain average of learned states. It will be shown later that these averaged states represent feasible power system states not presented during training. The mathematical properties of Kohonen networks, such as stability and convergency of the algorithm, are well established (Ritter and Schulten, 1988). Therefore, self-organizing networks present a promising approach for power system state classification.

The application of Kohonen networks for power system static security was proposed by us in 1991 (Niebur and Germond, 1991a). In the area of dynamic stability Kohonen networks were used for the estimation of the dynamic stability index by Mori, Tamaru and Tsuzuki, (1992).

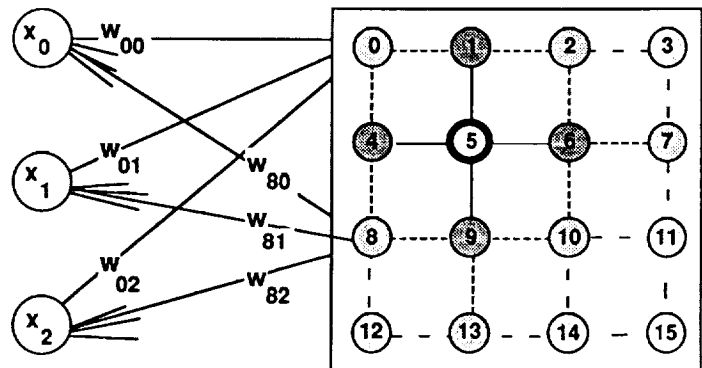
This report discusses an on-line decision aid for static security monitoring and assessment. Operating states are classified by using Kohonen's self-organizing feature map. We presented the general concept in 1991 (Niebur and Germond, 1991b). The next 2 sections give a short summary of the method and results. In the last sections the weight vector interpretation, choice of neural network parameters and security criteria are discussed.

SECTION 2

KOHONEN'S SELF-ORGANIZING FEATURE MAP

THE KOHONEN CLASSIFIER maps vectors of an N -dimensional space to a neural net of M neurons in a non-linear way, respecting the topological order of the vectors which, in general, is not known *a priori*. The M neurons are usually arranged on a 2-dimensional grid. Each input component is connected to each neuron by a weight w_{ij} ($0 \leq i < M$, $0 \leq j < N$). Each neuron i is therefore characterized by a weight vector w_i of the same dimension as the vector to be classified. Figure 1 shows the architecture of a 4×4 Kohonen network with 3-dimensional input and weight vectors.

Figure 1. Architecture of Kohonen's self-organizing feature map



Further, a neighborhood is specified for each neuron. If we define there to be at most 4 direct neighbors for each neuron, neuron 9, for example, has 4 neighbors of first order (neurons 5, 8, 10 and 13), 6 neighbors of second order (neurons 1, 4, 6, 11, 12, 14) and so on. The latter are all neighbors of first order of neuron 9's direct neighbors except neuron 9 itself. The neighborhood of higher orders are constructed recursively. Corner neurons (0, 3, 12 and 15) have only 2 neighbors of first order and border neurons (1, 2, 4, 7, 8, 11, 13, 14) have 3 first order neighbors.

Alternatively, if we define there to be 8 direct neighbors (arranged on the square around the neuron), then neuron 9 has 8 neighbors of first order (4, 5, 6, 8, 10, 12, 13, 14). All other neurons are neighbors of second order. Corner and border neurons have neighbors of order 1, 2, and 3.

The order of the neighborhood considered for neuron c during the training algorithm decreases monotonically with the time. It is defined by a neighborhood function $a(t, i, c) < 1$,

where t is the time, and c and i are the neurons whose neighborhood order is considered.

During the training phase the weight vectors are constructed as weighted sums of a certain number of training vectors. If the training vectors form a vector space (in the mathematical sense) the weight vectors are elements of the same space. They represent a set of training vectors which are situated in the same region of the vector space and which are therefore topologically close. Further, neighboring regions of the vector space are represented by the weight vectors of neighboring neurons. During the classification an input vector is classified by the neuron whose Euclidian distance between the weight vector and the input vector is the smallest.

Unsupervised Learning Algorithm:

- 1: $t := 0$: initialize all w_{ij} randomly
- 2: Choose input vector $\mathbf{x} \in X$ randomly in the training set
- 3: Determine the neuron c such that its weight vector \mathbf{w}_c is the closest to the input vector

$$\|\mathbf{w}_c(t) - \mathbf{x}\| = \min_i \|\mathbf{w}_i(t) - \mathbf{x}\| \text{ for all } i$$
- 4: Update the weight vector \mathbf{w}_c of the selected neuron c and the weight vectors \mathbf{w}_i of its neighbors

$$\mathbf{w}_i(t+1) := \mathbf{w}_i(t) + \alpha(t, i, c) (\mathbf{x} - \mathbf{w}_i(t))$$

$$\text{for } i = c \text{ or } i \in \text{neighborhood of } c$$

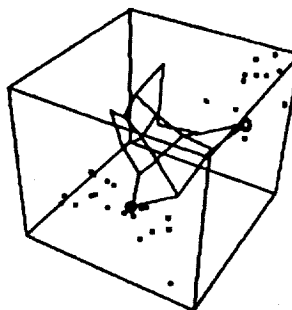
$$\mathbf{w}_i(t+1) := \mathbf{w}_i(t) \text{ elsewhere}$$
- 5: Increment the time $t := t+1$
- 6: if $\alpha(t, i, c) > \varepsilon$ goto 2 else STOP

Classification:

1. Get an input vector \mathbf{x}
2. Choose \mathbf{w}_c such that

$$\|\mathbf{w}_c(t_{\max}) - \mathbf{x}\| = \min_i \|\mathbf{w}_i(t_{\max}) - \mathbf{x}\| \text{ for all } i, 0 \leq i < M$$

Figure 2. Mapping of 3-dimensional vectors onto a 2-dimensional Kohonen network



In Figure 2, 500 3-dimensional input vectors are uniformly distributed in the volume of a 3-dimensional unit cube. They are mapped onto a 2-dimensional 4×4 Kohonen network. The 3-dimensional weight vectors will distribute regularly in the cube. Due to the small size of the network, the corner neurons will classify the extreme cases of vectors drawn near the boundaries

of the cube. For example the 2 clouds of points (input vectors) represented in Figure 2 are classified by the white and the black neuron, although these input vectors may not be close to each other in the sense of the Euclidian distance.

SECTION 3

MODELING OF THE SECURITY ANALYSIS PROBLEM

THE OPERATING POINT of a power system can be defined as a vector whose components are active and reactive line power measurements. According to the different security criteria, the boundaries of the secure state space are given by the operating constraints of the buses and lines. If the security criterion is prevention of line overloads, the boundaries of the secure domain of the state space are given by the maximum permissible currents of the transmission lines. Since operating points violating the same constraint are situated in the same region of the operating space, they will be classified either by the same neuron or by neighboring neurons. Therefore, secure operating points, ie vectors inside the boundaries of the secure domain, are mapped to a different region of the 2-dimensional neural net than insecure operating points. The dimension N of the operating space is $2 \times \text{number of lines}$.

Figure 3. Operating space of a 3-bus 3-line active power system model

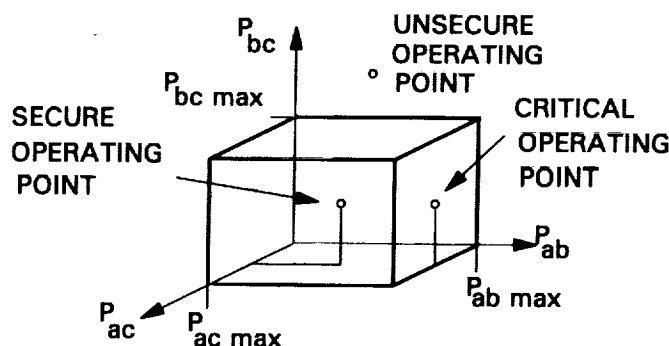


Figure 3 illustrates this concept for a 3 bus - 3 line active power system model. The small cube represents the secure part of the operating space. The external cube represents the outer borders of the insecure part of the operating space.

The cube in Figure 2 can now be interpreted as the operating space. The secure part of the operating space is not shown in Figure 2. If the neural net represents the operating space in the way illustrated, the border and corner neurons will classify the insecure cases whereas some of the inner neurons will classify secure and critical cases. The M weight vectors are therefore the prototypes of M classes of power system states.

It is not necessary for the classification concept that the security boundaries or operating space boundaries be defined by a cube. Since the Kohonen map classifies any 3-dimensional vector by the closest weight vector, points outside the cube will be classified by the border and

corner neurons of the illustrated network.

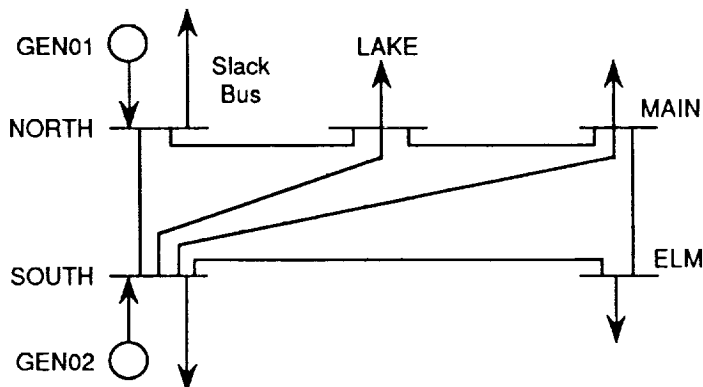
In the case of a non-convex secure space more neurons are needed to represent the more complex distribution of input vectors adequately.

In the following section these results are illustrated by a small example.

3.1 Example of a Five Bus Power System

The Kohonen network is used to classify line loading patterns resulting from single and double contingencies for a 5 bus - 7 line power system defined by Stagg and El-Abiad (1968) and represented in Figure 4.

Figure 4. 5-bus 7-line power system



Definition of the Input Vectors

For 7 lines the system state is defined by 7 complex components or, alternately, 14 real components, the 7 active and reactive line power flows obtained by off-line load-flow simulations. One load and generation scenario was defined for the base case. The training vectors were obtained by off-line load-flow simulations using the non-linear power system model. Different topologies of the network are generated by systematic $n-1$ and $n-2$ bus and line contingency simulations giving a set of 46 input vectors. A 5×5 neural network was trained with these vectors. The characteristics of the trained cases and the weight vectors are stored in a data base.

Reallocation of power on the generators was only performed for the outage of one generator, the missing power being reallocated to the other generator, or in some cases of double branch contingencies, where one or several loads cannot be supplied, the generator power was rescheduled.

In order to generate vectors which have not been trained but which are at a "reasonable" distance from the 46 trained vectors, load and generation of the base case were uniformly changed to 90%, 95%, 105% and 110%, providing a set of 184 untrained vectors.

3.2 Simulation Data and Performance

A simulator for the Kohonen learning algorithm was developed in the language C on an Apollo DN 3500 work station. In the simulator, vectors are drawn randomly from the set of 46 training vectors, and are presented as inputs to the self-organizing network. The neighborhood function

$\alpha(t, i, c)$ decreases exponentially in time t and in the order of neighborhood between neuron c and neuron i .

A significant reduction in training steps and therefore in training time can be achieved by initializing the weight vectors with an average of all input vectors and by then starting with a smaller neighborhood and a larger decay factor for the exponential function. (In fact, with an α near to 1 for a large number of steps and a big neighborhood, the network adapts each weight to the average of the input vectors in step 4 of the learning algorithm.)

For the 5×5 network, self-organization was satisfactory after 2000 instead of 3000 steps.

Experiments have shown that the number of direct neighbors (4 or 8) does not influence the quality of the results significantly. Training time is slightly higher (98 s instead of 67 s for the 5×5 network), and classification quality is slightly better for networks using 8 neighbors. (A 10×10 network with 8 direct neighbors needs 188 s for the 5 bus system. Training time for the IEEE 24 bus - 38 line system is about 60 min.)

SECTION 4

INTERPRETATION OF THE RESULTS

AT THE END of the unsupervised learning, the weight vectors are organized, and this organization reflects features of the power system which could not be easily determined, since it is difficult for a human being to interpret vectors in a N -dimensional space.

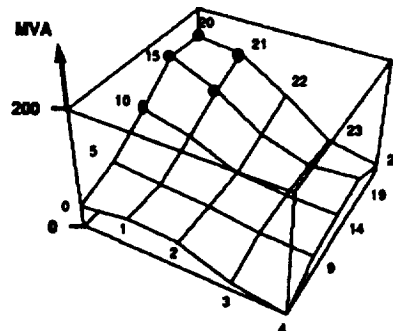
In the following, we try to extract the information from the organized weight vectors in terms of power system features using the map interpretation technique proposed by Tryba, Metzen and Goser (1988).

Each component of a weight vector represents a physical variable of the power system, an active or reactive branch power flow which is a synthesis of a certain number of input vectors.

For example, the first and second components of the weight vectors represent the active and reactive power in branch North-Lake. We determine from these two components the apparent power in MVA in branch North-Lake, and represent it in Figure 5 for each neuron. As defined in Figure 1, neurons occupy the edges of the squares and are numbered from 0 to 24.

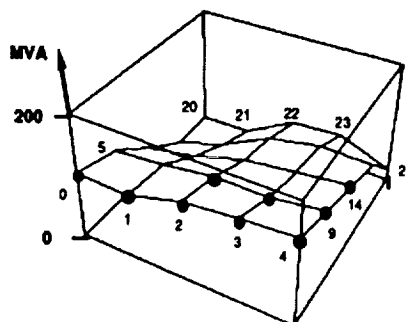
We can observe that the power system states stored in the weight vectors of neurons 10, 15, 16, 20 and 21 represent situations where the branch power limit is violated in branch North-Lake, assuming an acceptable limit of 100 MVA. This is indicated by black dots in Figure 5.

**Figure 5. Apparent power
in branch North-Lake**



Similarly, the interpretation of components 2 and 3, the active and reactive power in branch North-South, indicates that the apparent power limit represented in Figure 6 is violated in the power system states stored in neurons 0, 1, 2, 3, 4, 7, 8, 9 and 14. This is indicated by black dots in Figure 6.

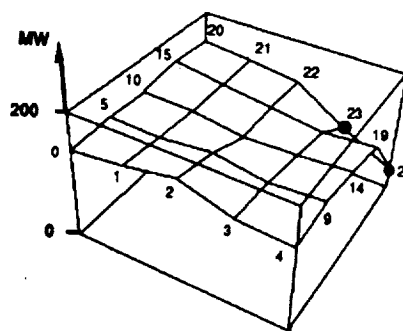
**Figure 6. Apparent power
in branch North-South**



Physical variables which were not contained in the input vectors can also be deduced, for example the active power generation at bus North, computed as the sum of components 0 and 2 of the weight vectors. This information, represented in Figure 7, indicates that the system load cannot be totally supplied in the states stored in neurons 23 and 24. The surface shown in that figure represents the total power at the swing bus. Therefore it is an indication of the total losses in the system, and the state stored in neuron 18 represents the state of the system with minimum losses.

The base case is mapped onto neuron 13.

**Figure 7. Active power generation
at bus North**



The other possible maps are interpreted similarly. Information is derived by evaluation of the weight vector components. Therefore the interpretation is valid for all neurons, whether they classify trained cases or not.

Finally the 5 bus voltages of all trained cases are looked up in the data base and neurons classifying cases with over or low voltage are identified. Neuron 5 and 0 classify cases with low voltage (< 85%) at bus Elm and neuron 8 low voltage at bus Main.

The synthesis of all the limit violations is shown in Figure 8, indicating the security boundaries with respect to line overloads, voltage magnitudes, and unsupplied load. Neurons are represented by the squares (and not as the edges of the squares as in Figure 7).

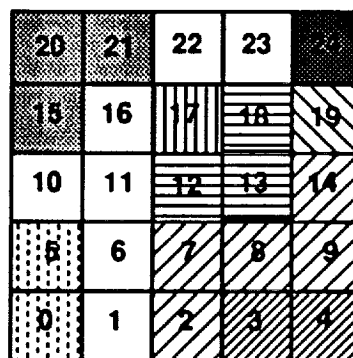
20	21	22	23	24
15	16	17	18	19
10	11	12	13	14
5	6	7	8	9
0	1	2	3	4











Figure 8. Security boundaries

Only system states classified by neurons 6, 11, 12, 13, 17, 18, 19 and 22 are secure. It is therefore possible to determine the nature of the unsecure states which are closest to a given state of the system, and to measure the distance to these unsecure states.

Figure 9 and its legend characterize the secure and unsecure regions in more detail with respect to the trained cases. The lines are labeled according to the first letters of the connected busses (see Figure 4).

Figure 9. 5×5 cluster map



-  N-S outage & N-L overload
-  N-S overload & outages other than N-L of type 1
-  N-L outage & N-S overload
-  N-S heavily loaded & outages other than N-L of type 1
-  Low voltage at E and heavy- or overload of line N-S
-  Isolated buses, unsupplied load
-  Cases without overload, base case and outages type 2
-  S-L + S-M outage, no overload
-  Neurons classifying no training vectors but some unknown cases
-  Neurons classifying no training vectors

type 1 = (Outages L-M or S-L and not(outage S-M or N-S or N-L))

or (N-S heavily loaded and outage S-L and not(outage S-M or N-S or N-L or S-E)).

type 2 = No overload and outage S-M and M-E and not(N-S or N-L or S-L or L-M or S-E).

The results show that the 5×5 network is well organized. A statistical analysis of the average distance and deviation between weight vectors and input vectors helps to determine the quality of the classification and identifies where misclassification can be expected. For example, neuron 14's components corresponding to the apparent power of N-S indicate 104 MVA, corresponding therefore to an overload. Since the average value of the input vector components is 106 MVA and the deviation 12.11 MVA, this neuron classifies heavily loaded cases as well as overloaded cases.

Neuron 13, on the other hand, indicates 94 MVA. For the input vectors classified by

neuron 13, N-S is loaded with 93 MVA on average with a deviation of 4 MVA. Trained cases classified by neuron 13 are critical but still on the safe side. This is no longer true for the classification of unknown cases where the new average, 98.12 MVA, and deviation, 10.05, indicate misclassifications.

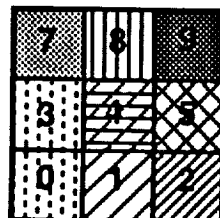
For the 184 untrained vectors only 6 overloaded cases are classified by neurons (12 or 13) which, according to Figure 8, indicate safe regions. Cases classified by neuron 1, 10 and 16 are known to be unsafe because of the weight vector analysis, although they do not classify any trained state.

SECTION 5

INFLUENCE OF NEURAL NET SIZE

THE INFLUENCE of the neural net size is discussed by examining the evolution of a 3×3 network to the 5×5 network. Figure 10 shows the results of the organization of the 3×3 network. The neurons have been renumbered in order to facilitate the comparison between Figure 9 and Figure 10. Figure 10 uses the same legend as Figure 9 plus 2 superposed squares for neuron 4 and 5. Neuron 5 merges all but 1 of the cases of neuron 14 and neuron 19, and neuron 4 merges the cases of neurons 7, 8, 12, 13, 17 and 18. Note that only neuron 8 classifies only secure cases. On the other hand, the overload cases for line N-S and N-L, the low voltage for bus Elm and the islanded system cases are still clearly distinguished.

Figure 10. 3×3 cluster map



$$\begin{array}{|c|} \hline \text{X} \\ \hline \end{array} = \begin{array}{|c|} \hline \diagup \\ \hline \end{array} + \begin{array}{|c|} \hline \diagdown \\ \hline \end{array}$$

= N-S heavily loaded or overloaded and outages other than N-L of type 1.

$$\begin{array}{|c|} \hline \text{X} \\ \hline \end{array} = \begin{array}{|c|} \hline \diagup \\ \hline \end{array} + \begin{array}{|c|} \hline \text{—} \\ \hline \end{array}$$

= N-S heavily loaded or overloaded, base case and outages type 2.

By augmenting the size of the neural network, some classes are divided into more specific subclasses. Further intermediate classes ("empty" neurons not classifying any trained vector) are created leading to a clear distinction of different groups of neurons called clusters. A 7×7 network, for example, already contains 24 empty neurons, 11 of which are used for the classification of unknown cases not fitting into any trained category.

The determination of the optimal system size and the scaling problem for real-world power systems will be addressed in future work.

SECTION 6

CHOICE OF SECURITY CRITERIA

POWER SYSTEM analysis and control treats system states under different security aspects. Data robustness, overload detection, voltage monitoring and contingency analysis are studied in security assessment. The proposed classifier is useful for all these analyses.

A strong feature of the self-organizing network is its robustness concerning incomplete or false data. Measurements containing **bad data** will simply be classified by the closest prototype state. Incomplete data can be handled by comparing known components with the corresponding weight vector component.

The information on the **bus power injections** is implicitly present in the input vector since the power of lines connecting to the same bus will add up to the bus power. This means that the input vectors lie on a manifold of the vector space. Therefore the actual dimension of the operating space is smaller than twice the number of lines.

The information on the **bus voltage** is present in the input vectors as a non-linear dependence (the usual load flow equation). These can be looked up in the data base for the trained cases.

In the case of a **contingency** of line a , for example, the components corresponding to the active and reactive power of line a are zero. The corresponding state vector lies in an $N-2$ dimensional subspace of the operating space. It can therefore be clearly distinguished from the base case except for the degenerated case where line a is so weakly loaded that the vector base case is situated close to the contingency vector. Another contingency would affect both cases similarly.

Assuming that neuron c classifies the base case and an unknown system state and neuron k the contingency of line a of the base case, then it can be concluded that

- 1) In general the base case and the unknown state display similar characteristics concerning line loadings, bus voltages and contingency consequences. These can be looked up in the data base.
- 2) In particular, an outage of line a will be classified by neuron k and therefore have the same consequences as outage a for the base case.

SECTION 7

CONCLUSIONS

THE CONTRIBUTION of this work is the application of Kohonen's self-organizing feature map to the classification of vectors representing power system states.

Each power system state to be learned is represented by an input vector formed of the active and reactive power flows in all the branches of the power system. Vectors of the training set, generated from load-flow simulations of a base case and contingencies, are presented as inputs to a Kohonen network, which uses unsupervised learning to adapt its synaptic weight vectors. Each weight vector corresponds to a neuron in the 2-dimensional Kohonen feature map.

To classify a power system state, whether it belongs to the training set or not, we present it as an input vector and determine the weight vector which is at minimal distance, and its associated neuron. Input vectors that are close together are assigned to neighboring neurons.

Since the weight vectors can be interpreted in terms of physical variables of the power system, the 2-dimensional feature map provides synthetic information about the states of an N -dimensional system.

A feature map of small size is already capable of forming clusters which exhibit significant information with respect to security analysis.

In a larger map, we observe that some of the neurons are associated with several training vectors, and some with none of the training vectors. However, the weight vectors for those "empty" neurons appear to represent feasible states of the power system, which can be understood as generalization from the learned vectors.

The network is capable of classifying cases which have not been learned. This feature is highly important for power system operation where it is unrealistic to expect that all possible cases will be encountered during off-line simulation.

Learning requires a significant amount of off-line computation to simulate the cases to be learned. The computation effort is mainly spent in the off-line load-flow simulations. The learning itself is presently done with a software program on a workstation. Dedicated processors are presently under study to improve the learning speed.

Note that the classification of a given input vector is extremely fast, even on a workstation, and could be performed in real time.

We are currently investigating further the behavior of this classifier for larger systems.

SECTION 8

REFERENCES

- Chow, J.-C., Fischl, R., Kam, M., Yan, H.H., and Ricciardi, S., (1990), "An Improved Hopfield Model for Power System Contingency Classification", Proceedings of IEEE-ISCAS, New Orleans, pp. 2925-2928.
- Ejebe, B. C., and Wollenberg, B. F., (1979, Jan/Feb), "Automatic Contingency Selection", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-98, No. 1.
- El-Sharkawi, M.A., Marks, R.J., Damborg, M.J., Atlas, L.E., Cohn, D.A., and Aggoune, M., (1990), "Artificial Neural Networks as Operating Aid for an On-line Static Security Assessment of Power Systems", Proceedings of the 10th PSCC, Graz, pp 895-901.
- Kohonen, T., (1989), "Self-Organization and Associative Memory", 3rd edition, Springer Verlag, Berlin.
- Mori, H., Tamaru, Y., and Tsuzuki, S., (1992, May), "An Artificial Neural Net Based Technique for Power System Dynamic Stability with the Kohonen Model", Proc. of PICA May 91, IEEE Trans. on Power Systems, Vol 7 No. 2, May 1992, pp. 856-864.
- Niebur, D., and Germond, A., (1991a), "Unsupervised Neural Net Classification of Power System Static Security States", Proc. of ESAP-91, Tokyo.
- Niebur, D., and Germond, A., (1991b), "Power System Static Security Assessment Using the Kohonen Neural Network Classifier", Proc. of PICA, May 91, IEEE Transactions on Power Systems, Vol 7 No. 2, May 1992, pp. 865-872.
- Oh, S. Y., (1986, Jan/Feb.), "A Pattern Recognition and Associative Memory Approach to Power System Security Assessment", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-16, No. 1, pp. 62-72.
- Pang, C. K., Prabhakara, F. S., El-Abiad, A. H., and Koivo, A. J., (1974, May/June), "Security Evaluation in Power Systems Using Pattern Recognition", IEEE Trans. on Power, Apparatus and Systems, Vol PAS-93 No. 2, pp. 969-976.
- Pao, Y.-H., (principal investigator EPRI EL-2343), (1982, April), "Feasibility of Using Associative Memories for Static Security Assessment of Power System Overloads", Project 1047-2, final report.
- Pao, Y.-H., (1990, April), "Applications of Neural-Net Computing to Power Systems", Proceedings of the NSF Workshop on Artificial Neural-Net Methodology in Power System Engineering, Clemson University, Clemson, SC.

- Ritter, H., and Schulten, K., (1988), "Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection", *Biological Cybernetics*, 60, pp 59-71.
- Sobajic, D. J., and Pao, Y.-H., (1989, Feb), "Artificial Neural-Net Based Dynamic Security Assessment of Electric Power Systems", *IEEE Transactions on Power Systems*, Vol 4, No. 1, pp. 220-228.
- Sobajic, D. J., Pao, Y.-H., Njo, W., and Dolce, J. L., (1990), "Real-Time Security Monitoring of Electric Power Systems using Parallel Associative Memories", *Proceedings of IEEE-ISCAS*, New Orleans, pp. 2929-2932.
- Stagg, G., and El-Abiad, A., (1968), "Computer Methods in Power System Analysis", McGraw-Hill, New York, p 284.
- Tryba, V., Metzen, S., and Goser, K., (1988), "Designing of Basic Integrated Circuits by Self-Organizing Feature Maps", *Proceedings of Neuro-Nimes*.

